

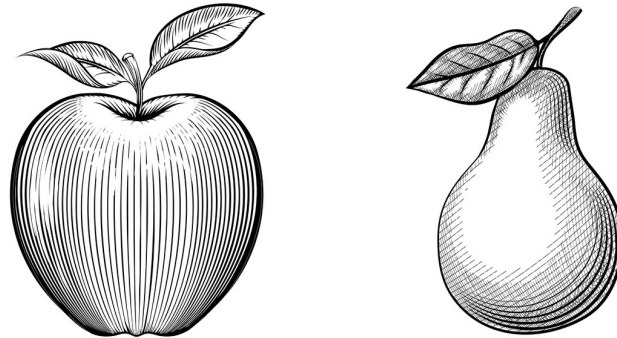
Programmeringsolympiaden 2021

TÄVLINGSREGLER FÖR SKOLKVALET

- Tävlingen äger rum på av skolan bestämt datum under **fyra timmar**. **Ingen förlängning ges för lunch eller raster**. Eleven ska i förväg komma överens med läraren om att använda egen dator eller en som skolan tillhandahåller. I vilket fall som helst måste eleven befinna sig i avtalad lokal på skolan. **På grund av pandemin kan i undantagsfall tävling ske från annan plats godkänd av läraren, givet tillräcklig övervakning**
- Tävlingen består av sex uppgifter som vardera ska lösas genom ett datorprogram i valfritt programmeringsspråk.
- **Indata kan läsas in i programmet på valfritt sätt**, t.ex. genom att programmet för en dialog med användaren (som i körningsexemplena i uppgifterna), att de skrivs in i ett grafiskt gränssnitt eller att datafiler skickas till *standard input*. Kom bara överens med din lärare om hur programmet ska testas.
- Dina lösningar kommer att testköras med förpreparerade indata. Varje uppgift testas normalt med 5 testfall, som vardera ger 1 poäng om ditt program skriver ut korrekt svar inom en exekveringstidsgräns av **3 sekunder**. Ingen test av indata behöver göras, den följer specifikationerna i uppgiften.
- Det är ofta olika begränsningar på de olika testfallen, t.ex. storleken på indata eller andra inskränkningar. Detta anges i uppgiften. **Observera att det kan vara helt olika svårighetsgrad på en uppgift beroende på dessa skillnader. Det kan därför vara lättare att få delpoäng på en uppgift som verkar svår än att få full poäng på en uppgift som verkar lättare**. Informationen om delpoäng är därför extremt viktig för att planera sin tävling.
- Rättningen utförs på samma eller likvärdig dator. Ändringar i källkoden tillåts ej efter tävlingen. Om programmet inte kan kompileras ges 0 p. på uppgiften.
- Om något av följande inträffar ger det *testfallet* 0 poäng, men programmet fortsätter testas med övriga testfall.
 - Exekveringstiden överstiger 3 sekunder
 - Exekveringsfel (run time error)
 - Fel svar
- Deltagandet är individuellt vilket bland annat innebär att inget utbyte av idéer eller filer får ske under tävlingen.
- Hjälpmedel: Valfritt skriftligt material, material som finns installerat på datorn samt material som finns tillgängligt på internet. Det är *inte* tillåtet att aktivt kommunicera på internet (t.ex. chatta eller ställa frågor till ett forum) utan endast att söka efter information. Räknedosa är tillåten.
- Tävlingsbidraget ska lämnas in i form av källkodsfiler (uppg1...uppg6 med passande filtillägg) som läggs på överenskommen plats. Ingen hänsyn tas till andra filer. Var noga med att lämna in den korrekta versionen av ditt program.

Lycka till!

UPPGIFT 1 – ÄPPLEN OCH PÄRON



Axel vill tävla med Petra om vem som sålt flest äpplen respektive päron, men Petra tycker inte att man kan “jämföra äpplen och päron”. De kommer överens om att de istället ska jämföra hur mycket de tjänat. De ber dig skapa ett program som, givet antalet äpplen Axel sålt och antalet päron Petra sålt, skriver ut vem som tjänat mest (eller “lika” om de sålt för lika mycket). Äpplen kostar 7 kronor styck och päron 13 kronor styck.

Körningsexempel 1

Äpplen ? 1
Päron ? 1

Svar: Petra

Körningsexempel 2

Äpplen ? 13
Päron ? 7

Svar: lika

Körningsexempel 3

Äpplen ? 15
Päron ? 8

Svar: Axel

UPPGIFT 2 – PERSONNUMMER

Svenska personnummer skrivs oftast med tio siffror på formatet ÅÅMMDD-XXXX. De sex första siffrorna utgör personens födelsedatum, så en person med personnummer 781113-3285 är t.ex. född den 13:e november 1978. En detalj som många inte känner till är att hundraplussare får ett plustecken istället för bindestreck i sitt personnummer. Detta är för att man inte ska blanda ihop dem med personer som föddes exakt 100 år senare. Till exempel kan någon som föddes år 1912 ha personnummret 121212+1212, medan 121212-1212 tillhör någon som föddes 2012.

Skriv ett program som läser in ett personnummer på formatet ovan, och skriver ut det på *tolvsiffrigt* format, d.v.s. ÅÅÅÅMMDDXXXX. Du kan anta att personerna i indatan föddes mellan 1840 och 2019. För enkelhets skull kommer det inte finnas några personer som föddes 1920.

Körningsexempel 1

Nummer ? 781113-3285

Svar: 197811133285

Körningsexempel 2

Nummer ? 121212-1212

Svar: 201212121212

Körningsexempel 3

Nummer ? 400506+1405

Svar: 184005061405

UPPGIFT 3 – HUNDRAELVA KRONOR

I Tumba pappersbruk — som är ansvariga för att producera sedlar — har tryckpressen gått sönder: den kan nu bara trycka siffran “1”. Att köpa en ny tryckpress kostar N kronor ($1 \leq N \leq 1\,000\,000\,000$), men pappersbruket har tyvärr helt slut på pengar. Det är ju dock de själva som trycker sedlar, så varför inte trycka nya pengar så att de kan köpa den nya maskinen?

Eftersom den trasiga tryckpressen bara kan trycka siffran “1” kan de endast trycka sedlar med valörerna 1 krona, 11 kronor, 111 kronor, 1111 kronor, o.s.v.

Pappersbruket undrar nu hur många sedlar de behöver trycka för att kunna betala för den nya tryckpressen. De vill kunna betala med jämna pengar, d.v.s. exakt N kronor (det är omoraliskt att trycka upp mer pengar än de behöver), och vill trycka så få sedlar som möjligt. Skriv ett program som beräknar antalet sedlar de måste trycka.

Poängsättning:

För 2 poäng gäller att $N \leq 1\,000$.

För ytterligare 1 poäng gäller att $N \leq 1\,000\,000$.

För de sista 2 poängen gäller att $N \leq 1\,000\,000\,000$.

Körningsexempel 1

Priset N ? 23

Antal sedlar: 3

Förklaring: Man använda en 1-kronasedel och två 11-kronorssedlar.

Körningsexempel 2

Priset N ? 12345

Antal sedlar: 5

Förklaring: Man kan använda en av varje av 1-, 11-, 111-, 1111-, 11111-kronorssedel.

Körningsexempel 3

Priset N ? 282828

Antal sedlar: 28

UPPGIFT 4 – LAVAPADDLING

Lav är tillfångatagen av en ond häxa i en vulkan. I det stora lavahavet i vulkanen ligger N öar på en lång rad ($1 \leq N \leq 20$). Avståndet mellan den i :te och den $i + 1$:te ön är d_i , och det gäller att $1 \leq d_1, d_2, \dots, d_{n-1} \leq 1000$. Avstånden är givna i enheten "häxameter", vilket motsvarar exakt H meter ($1 \leq H \leq 10^{12}$). Häxan bor på den första ön i raden, och det är här Lav befinner sig just nu.

Häxan har glömt sin bok med alla trollformler på den sista ön i raden, och tvingar nu Lav att åka och hämta den. Till sin hjälp har han en lavabåt och ett antal paddlar. Varje paddel håller för att ta K paddeltag ($1 \leq K \leq 15$), där varje paddeltag för båten framåt en meter, innan paddeln brunnit upp på grund av lavan. Lav får med sig ett antal paddlar från häxan, och behöver inte göra slut helt på en paddel innan han byter till en annan paddel.

Lav får dessutom med sig en trollformel från häxan som han kan använda när han står på en ö. Trollformeln lagar en paddel som han paddlat lite grann med utan att den brann upp helt. Då kan han återigen använda den för att paddla K meter. Lav kan använda trollformeln hur många gånger han vill när han står på en ö.

Skriv ett program som beräknar det minsta antalet paddlar häxan behöver ge Lav så att han kan klara uppdraget.

Poängsättning

För testfall värda 1 poäng gäller det att $K = 1$ och $H \leq 100$.

För testfall värda ytterligare 2 poäng gäller det att $H = 1$.

Körningsexempel 1

N ? 2
K ? 5
H ? 10
d1 ? 7

Svar: 14

Förklaring exempel 1:

Det finns två öar, med avstånd 7 häxameter, vilket motsvarar $7 \cdot H = 7 \cdot 10 = 70$ meter. Med varje paddel kan han ta max 5 paddeltag, så $70/5 = 14$ paddlar krävs.

Körningsexempel 2

N ? 3
K ? 7
H ? 100
d1 ? 2
d2 ? 1

Svar: 31

Körningsexempel 3

N ? 5
K ? 15
H ? 1000000000000
d1 ? 92
d2 ? 43
d3 ? 89
d4 ? 10

Svar: 6531851851852

Förklaring exempel 2:

Det finns tre öar, med 200 meter mellan den första och den andra, och 100 meter mellan den andra och den tredje. Varje paddel kan ta 7 paddeltag. Om Lav börjar med 31 paddlar, använder 14 av dem fullt, och tar 6 paddeltag med de resterande 17 paddlarna, så kommer han att ta sig $14 \cdot 7 + 6 \cdot 17 = 200$ meter, tillräckligt för att komma till andra ön. Då har han 17 paddlar kvar som han lagar med trollformeln, vilket räcker för att ta sig mellan andra och tredje ön. Det hade inte gått med färre än 31 paddlar.

UPPGIFT 5 – KLOCKAN

Timla åkte hemifrån vid midnatt och kom hem nästa midnatt, 24 timmar senare. Då hade det varit inbrott i huset! I huset har Timla en digital klocka som visar tiden i 24-timmarsformat med timmar, minuter och sekunder, alltid sex siffror. Timla gillar att spara el, så hennes klocka stänger av sig automatiskt när ingen är i huset. För att kunna mäta exakt hur mycket el hon sparar har hon också en mycket exakt elmätare. Enligt denna mätare förbrukade klockan N energienheter under dygnet då hon var bortrest. Varje siffersegment i klockan drar en energienhet för varje sekund som det siffersegmentet lyser. Siffrorna ser ut som på bilden. Exempelvis lyser 27 siffersegment då klockan är 02:41:35.



Timla vill hjälpa polisen genom att ta reda på vilken tid tjuven kan ha brutit sig in i huset. Man kan se på skoavtrycken att tjuven gick in i huset en gång och ut en gång, så klockan kan inte ha tänts och släckts flera gånger. Klockans display tänds och släcks alltid vid en hel sekund och kan som tidigast ha tänts 00:00:00 och kan som senast ha slocknat efter att ha visat 23:59:59. För givna N finns minst ett tidsintervall under dygnet då klockan förbrukar N energienheter.

Skriv ett program som beräknar antalet möjliga klockslag då displayen kan ha tänts.

Poängsättning

För 1 poäng gäller att $N \leq 23$.

För ytterligare 1 poäng gäller att $N \leq 200$.

Körningsexempel 1

Elförbrukning N ? 13

Antal möjligheter: 3

Förklaring: Klockan måste ha varit igång i exakt en sekund. Det kan ha varit under 11:11:17, 11:17:11 eller 17:11:11.

Körningsexempel 2

Elförbrukning N ? 2401920

Antal möjligheter: 1

Förklaring: Enda möjligheten är att klockan var på hela dygnet, från 00:00:00 till och med 23:59:59.

Körningsexempel 3

Elförbrukning N ? 100

Antal möjligheter: 3196

Förklaring: Det finns 3196 möjligheter för när klockan kan ha startat. Exempelvis kan den ha startat 20:02:06 och slocknat efter att ha visat 20:02:08.

UPPGIFT 6 – KODLÅS

Åskold har just köpt ett nytt kodlås. Försäljaren lovade att låset är mycket säkert, men Åskold är inte övertygad. Därför vill han att du ska räkna ut hur många olika kombinationer som öppnar låset.

Kodlåset består av N intelligande skivor. Varje skiva har M segment, där ett segment antingen är ifyllt eller ett hål. För att skriva in en kod vrider man på skivorna. Varje skiva kan sättas i M olika lägen, eftersom mekanismen inte tillåter att du vrider skivan mindre än ett helt segment. Låset öppnas om det *någonstans* går ett hål genom alla skivor på samma ställe.

Vi kan beskriva varje skiva som en sträng bestående av “.” och “#”, där “.” representerar ett segment med hål i och “#” representerar ett ifyllt segment. Att rotera en skiva ett steg kan då ses som att ta sista tecknet i strängen och lägga det i början. Om skivan roteras M steg kommer den tillbaka till läget den började i.

Exempelvis kan skivan “.#..#” ställas in i följande 5 lägen:

.#..# | #.#.. | .#.#. | ..#.# | #..#.

Totalt finns det alltså M^N möjliga sätt att ställa in de N skivorna, och låset öppnas ifall någon kolumn bara består av “.” när man skriver ut alla skivornas strängar ovanför varandra. Skriv ett program som beräknar hur många olika sätt det finns att ställa in skivorna så att låset öppnas.

Poängsättning:

För 2 poäng gäller att $M, N \leq 5$.

För ytterligare 1 poäng gäller att $N \leq 8, M \leq 10$ och att det finns max 3 hål i varje skiva.

För de sista 2 poängen gäller att $M, N \leq 12$.

Körningsexempel 1

Antal skivor N ? 2
 Antal segment M ? 3
 Skiva 1 ? .#. .
 Skiva 2 ? #..

Antal sätt: 9

Förklaring: Alla de nio inställningarna av skivorna låset.

Körningsexempel 2

Antal skivor N ? 3
 Antal segment M ? 4
 Skiva 1 ? ..#. .
 Skiva 2 ? ####
 Skiva 3 ? ..#. .

Antal sätt: 0

Förklaring: Ingen inställning av skivorna öppnar låset, då den andra skivan inte har några hål..

Fler körningsexempel finns på nästa sida.

Körningsexempel 3

Antal skivor N ? 3
Antal segment M ? 2
Skiva 1 ? #.
Skiva 2 ? .#
Skiva 3 ? #.

Antal sätt: 2

Förklaring: Följande två inställningar öppnar låset:

.#		#.
.#		#.
.#		#.

Körningsexempel 4

Antal skivor N ? 2
Antal segment M ? 3
Skiva 1 ? ..#
Skiva 2 ? .##

Antal sätt: 6

Förklaring: Det finns bara tre inställningar som *inte* öppnar låset: då “.” i det andra skivan är precis under “#” i den första skivan.